UNCLASSIFIED

DEFENCE R&D DÉFENSE

# Implementing and Testing a Real-Time Hierarchical Mine Detection Algorithm
## *(Final Report)*

Sinh Duong and Mabo R. Ito
University of British Columbia

Scientific Authority:
John McFee
Defence Research Establishment Suffield

**WARNING**

"The use of this information is permitted subject to recognition of propretary and patent rights."

**DEFENCE RESEARCH ESTABLISHMENT SUFFIELD**

Contract Report
DRES CR 2000-052

March 1999

National Defence Défense nationale

Canada

UNCLASSIFIED

102-00862/

UNCLASSIFIED

DEFENSE RESEARCH ESTABLISHMENT SUFFIELD
RALSTON, ALBERTA

DRES CONTRACT DSS FILE W7702-4-R486/01-XSG

# IMPLEMENTING AND TESTING A REAL-TIME HIERARCHICAL MINE DETECTION ALGORITHM

## FINAL REPORT

by

Sinh Duong and Mabo R. Ito

The University of British Columbia

Vancouver, B.C.

March 1999

UNCLASSIFIED

# Abstract

The Threat Detection Group (TDG) at the Defense Research Establishment Suffield (DRES) has undertaken a research program on the feasibility of the remote sensing of minefields. One of the projects is the Remote Minefield Detection (RMD) hierarchical algorithm originally developed for single band infrared airborne imagery.

Recently, in order to reduce the serious problem associated with the mining of important supply/communication roads by hostile parties during peace keeping operations, the Department of National Defense has begun the development of a multi-sensor remote mine detection vehicle. This idea leads to the necessity to scale down the Remote Minefield Detection system to support ground based operations with real time performance. The objectives of this contract are to implement the hierarchical mine detection algorithm developed under previous contracts on a down-scaled version of the target system for both real-time and non real-time operation, and to test the system on images obtained from a number of sources such as active airborne infrared imagers (AAIR), hyper spectral imagers (CASI) and vehicle mounted passive infrared cameras (FLIR).

This Final Report will first focus on the implementation of a real-time RMD system on a transputer-workstation network to detect mines in synthetic AAIR, CASI and FLIR images. Then it will report the development of a non real-time simulation of the algorithm on a Unix machine for detecting mines in real FLIR images. Finally, suggestions for future development will be discussed, and a brief introduction of expert system and its applicability to the project will be included.

# Table of Contents

1. INTRODUCTION ..... 7

2. LAND MINES AND RMD ALGORITHM ..... 10

2.1 Land Mines and Mine Detectors ..... 10
2.1.1 Airborne Active Infrared Imager (AAIR) ..... 10
2.1.2 Compact Airborne Spectrographic Imager (CASI) ..... 12
2.1.3 Forward Looking Infrared (FLIR) ..... 13
2.2 RMD Hierarchical Algorithm ..... 15

3. REAL-TIME DOWN-SCALED RMD SYSTEM ..... 17

3.1 Software Development ..... 17
3.2 Hardware for Down-scaled RMD System ..... 18
3.3 Preliminary Test Results on Down-Scaled System ..... 25

4. MINE DETECTION IN FLIR IMAGES ..... 28

4.1 FLIR Imagery Characteristics ..... 28
4.2 Algorithm Modification and Implementation for FLIR imagery ..... 28
4.2.1 Non-Suspect Region Rejection (NSRR) ..... 29
4.2.2 Local Region Thinning (LRT) ..... 30
4.2.3 Local Region Segmentation (LRS) ..... 31
4.2.4 Local Region Feature Extraction (LRFE) ..... 33
4.2.5 Local Region Classification (LRC) ..... 34
4.3 Implementing and Testing the RMD-FLIR System: ..... 37

5. CONCLUSION ..... 40

APPENDIX A: EXPERT SYSTEM AND ITS APPLICABILITY ..... 42

A.1 Expert System Shell ..... 42
A.2 Scope of the Knowledge Base ..... 43
A.3 Testing Methodology ..... 44

APPENDIX B: FEATURE SELECTION FOR OBJECT IDENTIFICATION ..... 45

APPENDIX C: COMPUTATION OF PROCESSING TIME FOR NSRR, LRT AND LRFE ..... 48

C.1 Computation of processing time of the NSRR module ..... 48
C.2 Computation of processing time of the LRT module ..... 50
C.3 Computation of processing time of the LRFE module ..... 51

APPENDIX D: RUNNING FLIR HIERARCHICAL ALGORITHM SIMULATION ..... 54

REFERENCES ..... 58

# Glossary

**Airborne Active Infrared (AAIR):** Active infrared images obtained from monochromatic airborne imagers.

**Compact Airborne Spectrographic Imager (CASI):** A pushbroom VNIR hyperspectral scanner which requires a smooth linear forward motion, normally that of an aircraft containing the instrument, to produce a two dimensional spectral image.

**Forward Looking Infrared (FLIR):** Passive infrared imagery obtained from cameras mounted on mine detecting vehicle.

**Global Region Classification (GRC):** The stage in the high level of the Remote Minefield Detection hierarchical algorithm that assigns the feature vectors to their most likely group.

**Global Region Feature Extraction (GRFE):** The stage in the high level of the Remote Minefield Detection hierarchical algorithm in which various morphological features such as pixel area, average pixel intensity and region compactness are measured for image regions.

**Image Correction (IC):** The stage in the Remote Minefield Detection hierarchy that compensates for distortions, dropouts, overlapping swaths, misregistration, and other artifacts and imperfections due to the scanning process.

**Improved Landmine Detector Project (ILDP):** A project to provide a remotely operated multi-sensor, vehicle mounted mine detection system to detect and mark low-metal content landmines.

**Local Region Classification (LRC):** The stage in the low level of the Remote Minefield Detection hierarchy that assigns the feature vectors to their most likely group.

**Local Region Feature Extraction (LRFE):** The stage in the low level of the Remote Minefield Detection hierarchy in which various morphological features such as pixel area, average pixel intensity and region compactness are measured for image regions.

**Local Region Assemble (LRASSEMBLE):** An extension of LRT module to construct grey scale subimages around the suspect objects so that LRS can analyze.

**Local Region Segmentation (LRS):** The stage in the low level of the Remote Minefield Detection hierarchy which segments the subimages formed by the LRT stage into regions.

**Local Region Thinning (LRT):** The stage in the Remote Minefield Detection hierarchy that eliminates redundant subimages formed by the NSRR stage.

**Non-Suspect Region Rejection (NSRR):** The stage in the Remote Minefield Detection hierarchy which reduces the image data to small subimages that are likely candidates to contain mines.

**Remote Minefield Detection (RMD):** The process of detecting minefields from an airborne platform. Also called Standoff Minefield Detection.

**Remote Minefield Detection Hierarchy (RMDH):** A scheme which includes a collection of algorithms for the purpose of detecting and identifying minefields from images.

**Threat Detection Group (TDG):** A research group at the Defense Research Establishment Suffield working on the problem of detecting ordnance such as landmines and artillery shells.

**Target Spatial Analysis (TSA):** The stage in the Remote Minefield Detection hierarchical algorithm which attempts to describe and interpret the relative positions of suspect mines.

# 1. Introduction

The Threat Detection Group (TDG) at the Defense Research Establishment Suffield (DRES) has been investigating real-time methods of detecting minefields from airborne imaging platforms and close-in detection of scatterable and patterned minefields, commonly called Remote Minefield Detection (RMD).

The University of British Columbia (Chief investigator: Prof. Mabo Ito), under contract and in collaboration with DRES for the last several years, has been studying methods of processing airborne minefield images in real-time with an aim toward detecting mainly surface-laid minefields. This has led to a hierarchical image analysis algorithm for single band (monochromatic) imagery that has the potential for real-time detection of minefields.

The hierarchical algorithm, described in [1] and illustrated using Figure 2.4, consists of the following levels: Preprocessing; Target Cueing (Non Suspect Region Rejection, Local Thinning); Target Shape Analysis (Local Region Segmentation, Local Region Feature Extraction, Local Region Classification); Target Spatial Analysis (Syntactic Classification, Statistical Classification); Knowledge Integration (Syntactic Classification, Data Fusion).

Under the contract (DSS W7702-6-2795), the framework and most of the low level parts of algorithm was developed and implemented on a general purpose vector processor-based image analysis system. The second contract (DSS W7702-9-R1211) initiated the study of the feasibility of real-time operation. The redesigned system hardware was based on an array of vector processors communicating with each other and a network of transputers through an I/O computer (IOC). Middle levels of the hierarchy were developed, and the existing parts of the algorithm were reformulated to execute on the new architecture. The third contract (DSS W7702-1-R286) developed the upper level of hierarchy. The algorithm was redesigned to use transputer based vector processors and eliminate the IOC; data and message passing protocols were resolved. An expert system with rudimentary rule set and knowledge base was also installed and debugged.

Since the research began, it has become apparent that the algorithm might have applications other than the single band infrared airborne imagery for which it was originally intended. For example, recent work has shown that hyperspectral images of surface-laid mines obtained from a Compact Airborne Spectrographic Imager (CASI) can be reduced to single band high contrast images which are very amenable to analysis by the algorithm.

Also, the Improved Landmine Detector Project (ILDP), which is a very high priority project within the Defence Research and Development Branch (DRDB), intends to develop a vehicle mounted mine detector for use on roads. Multiple detectors will be employed with data fusion being used to reduce false alarms. A passive infrared imager is a candidate detector for the system. A simple automatic target recognition (ATR) algorithm is being developed to assist the primary operator-in-the-loop detector. Preliminary images show that a simple detection

algorithm may yield a poor probability of detection (PD) and a high false alarm rate (FAR). The hierarchical minefield detection algorithm is inherently more robust and may, if adapted to the vehicle-mounted problem, provide better results. It is thus desirable to adapt the RMD algorithm as appropriate for use on CASI and ILDP (FLIR) images and test it on images acquired from both systems.

The objectives of this contract (DSS W7702-4-R486) are to complete the implementation of the hierarchical mine detection algorithm on a transputer-workstation computer system for real-time operation, and to test this system on images obtained from a number of sources such as airborne active infrared imagers (AAIR), hyperspectral imagers (CASI) and vehicle mounted passive infrared cameras (FLIR). However, due to the unavailability of real AAIR and CASI minefield images at the time and a R&D staff reduction, this contract will emphasize mainly on a real-time down-scaled system to detect mines from synthetic AAIR images, and a simulation version of the RMD algorithm to detect mines from real FLIR images. The first goal signifies that the algorithm can be actuated on a hardware network to serve in real-time as intended. The second task is more important because it proves that a RMD implementation will work well with real image data despite of unfavorable factors such as clutter or distortion, and it helps as a guideline for algorithm modification to adapt to other types of imageries.

As part of the contract, an expert system was planned to handle the high level of the algorithm. Although a testing methodology and a scope of the knowledge base had been defined to verify that the expert system would work correctly, the knowledge base has only been implemented in a cursory manner. Detailed knowledge base engineering, system integration, and testing the complete expert system has not been done.

This contract has three phases. The first phase consists of the following tasks:

1. Implement a complete RMD algorithm on a real-time down-scaled system consisting of array processor(s) and T800 TRAMs, which connect to a Sparcstation.

2. Test this system on available Remote Minefield Detection imagery, i.e., synthetic AAIR images.

3. Estimate hardware needed for complete target system for remote sensed minefield imagery (AAIR), vehicle mounted mine detector passive infrared imaging (FLIR), and hyperspectral imagers (CASI).

4. Acquire target system hardware equipment.

5. Acquire ground based passive infrared images (FLIR).

The second phase worked on:

1. Adapt the RMD algorithm for vehicle mounted IR mine detection (FLIR).

2. Implement the RMD for FLIR as a non real-time system, up and including the Local Region Feature Extraction (LRFE) stage, to detect mines in real FLIR images. This simulation version runs on a Sparcstation computer.

3. A introductory study of expert system and its feasible applicability to the RMD project.

The third phase was modified after discussions with the scientific authority. Due to a lack of appropriate CASI minefield imagery, it was mutually agreed to not do items a-h of phase 3 of the Statement of Work, i.e., non real-time RMD system for CASI, data fusion and expert system knowledge base. Instead, full works concentrated on:

1. Complete the non real-time RMD system for FLIR.

2. Test and tune the overall system using single-frame single-band real FLIR images.

3. Suggest of enhancements for future development.

# 2. Land Mines and RMD Algorithm

This section briefly introduces land mine and mine sensor characteristics, then it describes the structure of the Remote Minefield Detection hierarchical algorithm.

## 2.1 Land Mines and Mine Detectors

Land mines are often cylindrical but may be a variety of shapes.[2] Diameters range typically from 10 to 30 cm for anti-personnel and anti-tank mines, respectively. Outer cases vary in spectral reflective characteristics and materials depending on where they are deployed and include bare metal, plastic, paint or wood.

Minefields may consist of buried or surface-laid mines. The former are slowly laid by mechanical means and may be detected by looking for patterns of soil or vegetation disturbance immediately above the mines or disturbances caused by the mine laying machinery. Surface-laid (scatterable) mines may be dispensed by land vehicles or delivered by artillery or helicopter in patterns or random distributions. It is such minefields which are of particular interest in our work. Air delivery may leave no disturbance due to the delivery except, in some cases, for parachutes or dispensing containers. This makes air-delivered scatterable mines a serious threat. A typical minefield distribution might consist of over 200 mines in a 500 m square area.

Space allows only a brief description of the scenarios and constraints on the problem.[1] For most minefield images the number of targets (i.e. mine-like objects) is few and the area of each is small compared to the total search area. Hence, a typical image will be dominated by non-target areas. To be useful a sensor must be able be show a significant number of compact regions of particular shapes, which could be mine-like objects and which contrast significantly with the background. Also, such objects normally will be sparsely distributed over a large area. In practice, it is usually necessary that each target be at least several pixels wide to ensure adequate shape discrimination.

Since the search area for mine-like objects must necessarily be large, input data rates can also be very large if real-time detection is required. As a typical example, an airborne active infrared imager can have an input data rate of over 8 Mbytes/s. The ratio of total target area to total search area is roughly 1:25000, and a target may have between 50 and 250 pixels.[3]

A number of imaging detectors are being developed for use in remote minefield detection from airborne platforms and slow moving ground vehicles and are briefly described below.

### 2.1.1 Airborne Active Infrared Imager (AAIR)

This active infrared imaging system[17] can produce single-band, high spatial-resolution imagery in which mines contrast significantly with the background. In monochromatic airborne imagery, such as shown in Figure 2.1, a mine appears as a few pixel wide region. A small anti-personnel mine appears as about 8 pixels in diameter and covering ~ 50 pixels square area. On the other hand, a large anti-tank mine is ~16 pixels in diameter and covers ~ 215 pixels square area. The minefield is sparsely distributed over a large area resulting in a high input data rate from the scanner. For instance, a typical scenario is an aircraft flying 220 km/h at a nominal altitude of 150 m and equipped with a 10.6 micrometer wavelength active laser ($CO_2$) imager with an angular resolution of 0.1 mili-radian. This imager can generate a spatial resolution of 1.5 cm square, 2048 samples/line (a swath width of 30.72m) and at a data rate of 4096 lines/s. At 1 byte/pixel this yields a data rate of about 8.4 Mbytes/s.

**Figure 2.1:** An airborne image with pattern minefield.

## 2.1.2 Compact Airborne Spectrographic Imager (CASI)

This visible wavelength imaging sensor[4, 18] generates multi-spectral radiometric images of mines and back grounds which are converted to reflectance images using an incident light sensor. Mine-like objects are contrasted with the background by processing the reflectance spectra in two ways. The first discriminates individual pixels using the linear correlation coefficient as a measure of spectral similarity while the second discriminates spectra using a variant of linear spectral un-mixing in which the majority spectral members within an image are treated as background. Comparison of the two methods revealed that the correlation method is better for high spatial resolution data. When the targets were sub-pixel in size, the spectral member analysis had a higher probability of detection than the correlation method, but had more false alarms.

CASI is a pushbroom scanner which requires a smooth linear forward motion, normally that of the aircraft containing the instrument, to produce a two dimensional spectral image. Initially CASI was tested on a stable ground-based platform which produced high resolution imagery. In these high resolution images of replica mines, targets were discriminated from a variety of background types, even when partially obscured by vegetation, for widely varying illuminations caused by diurnal and seasonal variations, sky conditions and sun angles. Recently in preliminary practical tests, the CASI was flown over various areas, such as agricultural fields, as shown in Figure 2.2, in which sub-pixel-size mine-like targets were laid. Visually undetectable targets were detected with good results.

The hyperspectral imager has up to 288 bands each 4 nm wide in wavelength nominally covering the region 400 to 1000 nm wavelength. The imager is a pushbroom scanner with up to 512 samples per line and as many lines as one choose to collect. Raw data must be corrected for electronic offset, dark current, scattered light and frame shift smear, then converted to spectral radiance units. The multiband images are next reduced to a single band by some techniques, one of which is the linear correlation coefficients calculation.[4, 18] In some configurations, the data rate output from the imager is about 1 Mbyte/s or 81.4 lines/s for 512 samples/line, and the imager utilizes 16 bands of 12 bit data. After pre-processing, a stream of single band images is fed into the Remote Mine Detection system at rate of 20 lines/s. A large anti-tank mine is typically 5 pixels along and across track, and a small anti-personnel mine is about 1 pixel.

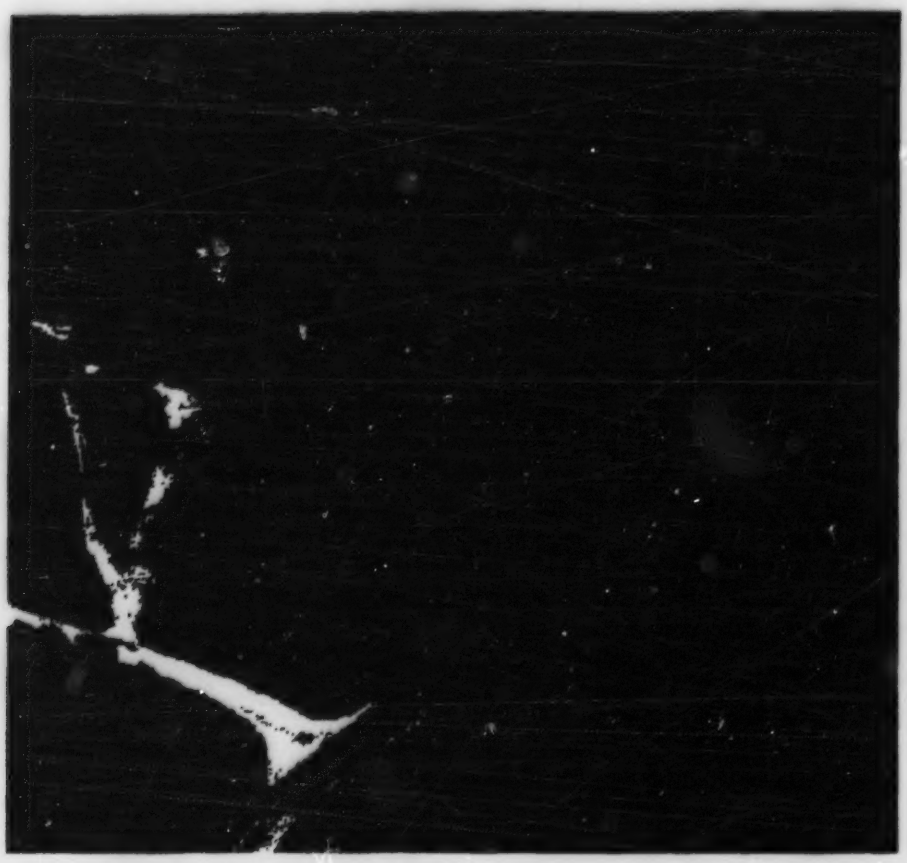


**Figure 2.2:** A CASI image.

In Figure 2.2, the image at left is an RGB image obtained from overflight of a mined area. After processing the CASI image, the single pixel or subpixel-size mines appear in patterns as shown at right.

## 2.1.3 Forward Looking Infrared (FLIR)

A multi-sensor remote mine detection vehicle will operate primarily on roads at slow speed about 3.6 km/h.[5] One sensor identified as a serious instrument for mine detection is the passive long wave ( 8 - 12 micrometer wavelength) infrared camera because of its low-metal mine detection capabilities. During recent years, Canadian defence scientists evaluated this technology as a function of the working scenarios. Different trials have been designed to evaluate the probability of detection and the false alarm rate of that technology as a function of many parameters such as:

time of day and year, weather conditions, type of road, geographic location, and type of mine. In addition, the mine detection efficiency of human perception in a simulated environment and different image contrast enhancing algorithms have been analyzed.

A typical data rate for this imagery is 5 images/s or 2000 lines/s. An image has 400 lines, each of which contains 800 16-bit samples. Due to the range dependent perspective, the size of mines depends on their position in the frame. In mid frame range, an anti-tank mine may be 32 - 40 pixels in diameter and an anti-personnel may be 12 - 20 pixels in diameter, along track and across track respectively. Typical expected signal to noise ratio are in the range 10 to 50. Figure 2.3 shows a typical FLIR image from a vehicle mounted camera.

**Figure 2.3:** A FLIR image with visible anti-tank and anti-personnel mines.

## 2.2 RMD Hierarchical Algorithm

The structure of the Remote Minefield Detection algorithm[1] for real-time detection of sparse small objects in images can be described using Figure 2.4. The major parts of the algorithm consists of the Low Level Target Cueing to reject non-suspect regions and thus drastically reduce the data rate, the Middle Level Target Shape Analysis to classify the suspect regions as target or non-target relying upon their morphological features, the High Level Target Spatial Analysis to extract the features of the spatial relationship between mine-like objects, and the Top Level Knowledge Integration to resolve whether or not the image contain mines using the spatial analysis results and external information resources.

Raw image data is acquired by a sensor and passed into the Image Correction (IC) stage. This IC module adjusts the raw image data to compensate for distortions, dropouts, overlapping swaths, misregistration, and other artifacts and imperfections due to the scanning process. After the IC stage, the data is passed to the Non-Suspect Region Rejection (NSRR) stage. The NSRR reduces the immense data flow down to a stream of small images (subimages) that are likely candidates to contain mines. To accomplish this task, the NSSR collects a block of scan lines of the image and then divides it into smaller non-overlapping square regions called subimages. Each subimage is further divided into smaller non-overlapping square regions, whose width is smaller than the dimensions of a small target but large enough to provide a stable average of the region. The contrast values of these small regions against the parent subimage are calculated and ranked. Only a few top values will be selected as suspect regions. Then the data of these suspect regions is transferred to the Local Region Thinning (LRT) module where redundant subimages are eliminated. Within this module, function Local Region Assemble (LRASSEMBLE) constructs subimages around the suspect objects. The next stage, Local Region Segmentation (LRS), partitions these subimages into homogeneous regions. These regions are then passed to the Local Region Feature Extraction (LRFE) block. In this step, various morphological features such as pixel area, average pixel intensity and region compactness are measured for each candidate mine-like object. The next stage, Local Region Classification (LRC), categorizes the assembled feature vectors into different classes, to determine targets or non-targets with an estimated likelihood based on extracted features. The results of the classification are passed to the higher level, Target Spatial Analysis. Here, initially the relative positions of likely mines are analyzed and various spatial size and shape clusters are formed using a Clustering algorithm. Then, the clusters are delivered to Global Region Feature Extraction (GRFE) where both statistical measurements and pattern descriptors of each cluster, as well as spatial inter-relationship among clusters, are computed and extracted. Depending on the type of patterns encountered, either a statistical or a syntactic pattern classifier may be more appropriate. The positions of mines in scatterable minefields can be well described by probability density functions and hence are amenable to statistical classification. The positions of mines in patterned minefields are better described by deterministic functions, which suggests syntactic classification. However, since the patterns are not known in advance, both classifiers are needed and operate in parallel. In the final level, Knowledge Integration, the expert system integrates the statistical and syntactical data output from GRFE with other sources from the user and knowledge base to arrive at a decision

on whether or not the raw image data likely contains a minefield. More details of the real-time mine detection hierarchical algorithm are given in previous papers.[2, 3, 16]

This system architecture was initially designed for AAIR airborne imagery. Some modifications of the algorithm and hardware structure are being investigated to accommodate CASI and FLIR situations.

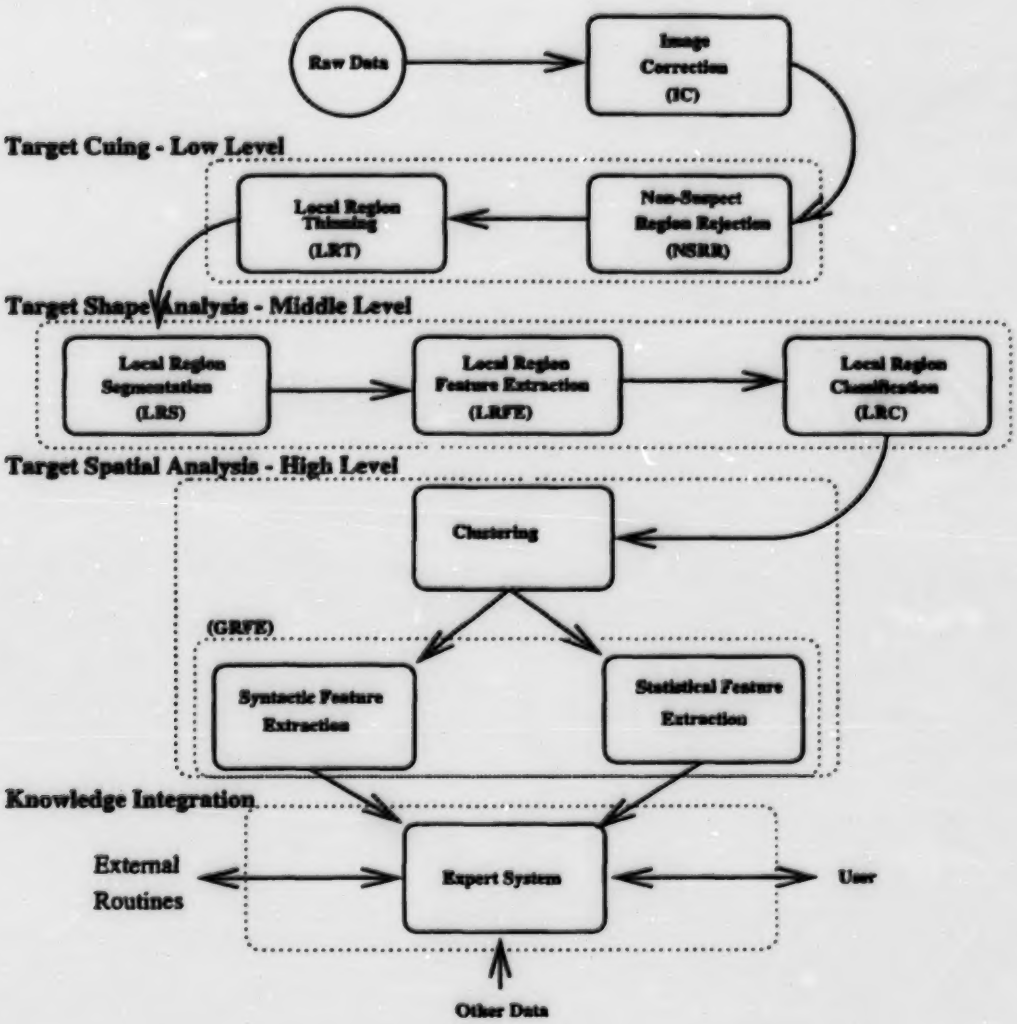



**Figure 2.4:** Structure of the Remote Minefield Detection hierarchical algorithm.

# 3. Real-time Down-scaled RMD System

In phase 1, the RMD hierarchical algorithm was implemented on a down-scaled version of the target system. The propose of this work is to size the amount of program code needed and to estimate the amount of parallel hardware required to get real-time performance.

## 3.1 Software Development

Discussion of software development of a complete RMD algorithm is given in detail in the report "Image Analysis Application Development" [2]. A non real-time version of the low and middle level of the hierarchy was first programmed to run on a Unix-Sparc machine, and it took synthetic AAIR images as input test data. This simulation system's codes were located in the following TDG Sparcstation's directory:

RMD simulation version for airborne imagery
programmers: J.E. McFee, K.L. Russell, R.Q. Vu
location: /net/frodo/export/home/projects/rmd/hieralgorithm/lowmiddlesim
language: IDL, C

After verifying the non real-time program running as intended, the code of this low and middle level of RMD hierarchical algorithm was transferred to implement on a real-time down-scaled version of the target system, utilizing a network of vector processors i860s and transputers TRAMs. The preliminary test result of this version is reported in section 3.3. The software was stored in directories:

- NSRR and LRT modules
  programmers: R.Q. Vu, S. Duong
  location: /net/frodo/export/home/projects/rmd/hieralgorithm/nsrr_lrt
  language: IDL, C, OCCAM

- LRS module
  programmer: K.L. Russell
  location: /net/frodo/export/home/projects/rmd/hieralgorithm/lrs
  language:

- LRFE module
  programmer: R.Q. Vu
  location: /net/frodo/export/home/projects/rmd/hieralgorithm/lrfe
  language: OCCAM

- LRC module
  programmer: J.E. McFee
  location: /net/frodo/export/home/projects/rmd/hieralgorithm/lrc
  language: C and OCCAM

Samples of minefield images were stored in the following directories:

- Synthetic AAIR images
  location: /net/frodo/export/home/projects/rmd/images/idl/minefield/simulation/data
  format: grey scale binary

- CASI images
  location: /net/frodo/export/home/projects/rmd/images/casi
  format: PCI pix

- FLIR images
  location: /net/frodo/export/home/projects/rmd/images/flir
  format: AGEMA's image format

## 3.2 Hardware for Down-scaled RMD System

**3.2.1 Systems Integration Analysis:** Because of the immense data flow present in the low and middle levels of the RMD hierarchy, it is essential to study the systems integration of the modules, to resolve difficulties in communication between the various modules of the algorithm and to determine the optimal hardware configuration. First, the high input data rate suggests a simple, high speed Target Cueing (described in page 16), which does not identify target objects by their shapes. Image understanding techniques can do shape recognition and spatial relationship analysis but they cannot tolerate the high raw data input rates. Since it is desirable to process the imagery in real-time, a fusion of the two methods is necessary. One approach is to use a hierarchy of processing levels which pass data from one level to the next. At the higher levels, computational complexity increases, but the volume and rate of data flow decrease.

Further reading on the systems integration analysis of the RMD algorithm on a network of transputers can be found in the previous contract reports [3] and [16]. Figures 3.1, 3.2 and 3.3 illustrate the data flow analysis performed on AAIR, CASI and FLIR imageries, respectively. Note that because FLIR images are obtained from ground based vehicles which are intended to detect individual mines, there is no spatial analysis required to detect a whole minefield.

Note also that real remote minefield detection imageries CASI and FLIR were not yet available at the time of this analysis. Thus parts of a synthetic AAIR image (1024x1024 pixels) were carefully cropped, accordingly to the standard sizes of CASI (512x512 pixels) and FLIR

(800x400 pixels), to resemble these images. Identical RMD algorithms were applied for these three AAIR, CASI and FLIR images; and therefore many data results, such as subimage size and threshold Iclip in NSRR or vector size in LRFE, which effect the data flow are the same. In future, real CASI and FLIR images might require slightly modified algorithms, and thus a new systems integration analysis will be needed.

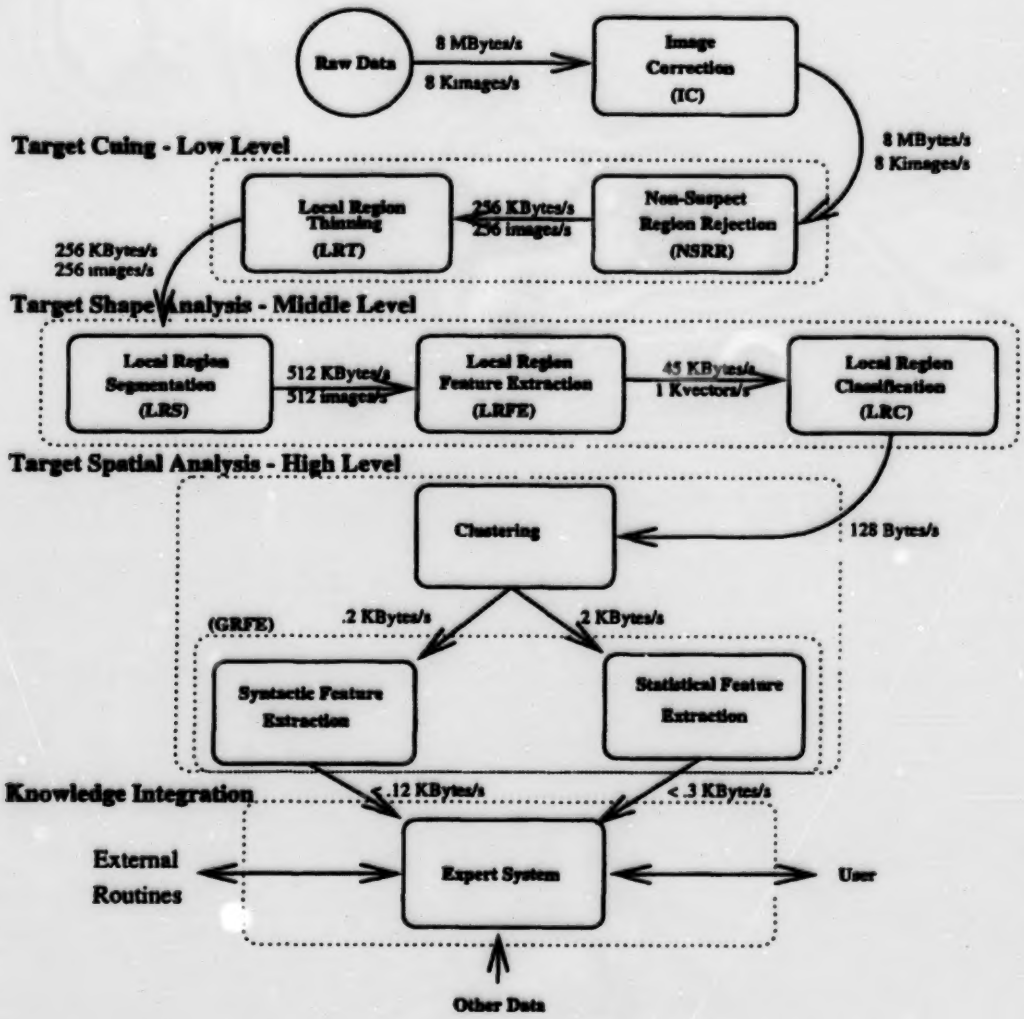



**Figure 3.1:** Systems integration analysis for AAIR imagery.

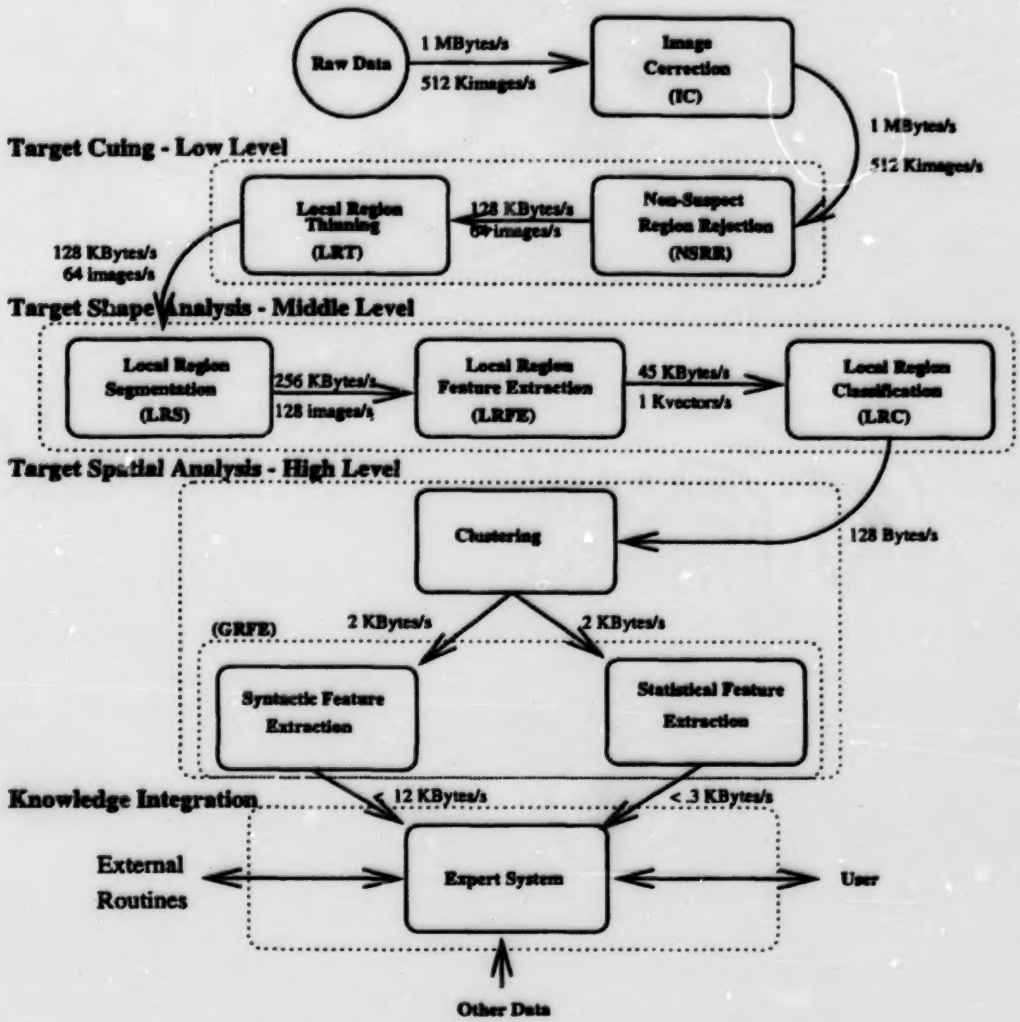



**Figure 3.2:** Systems integration analysis for CASI imagery.

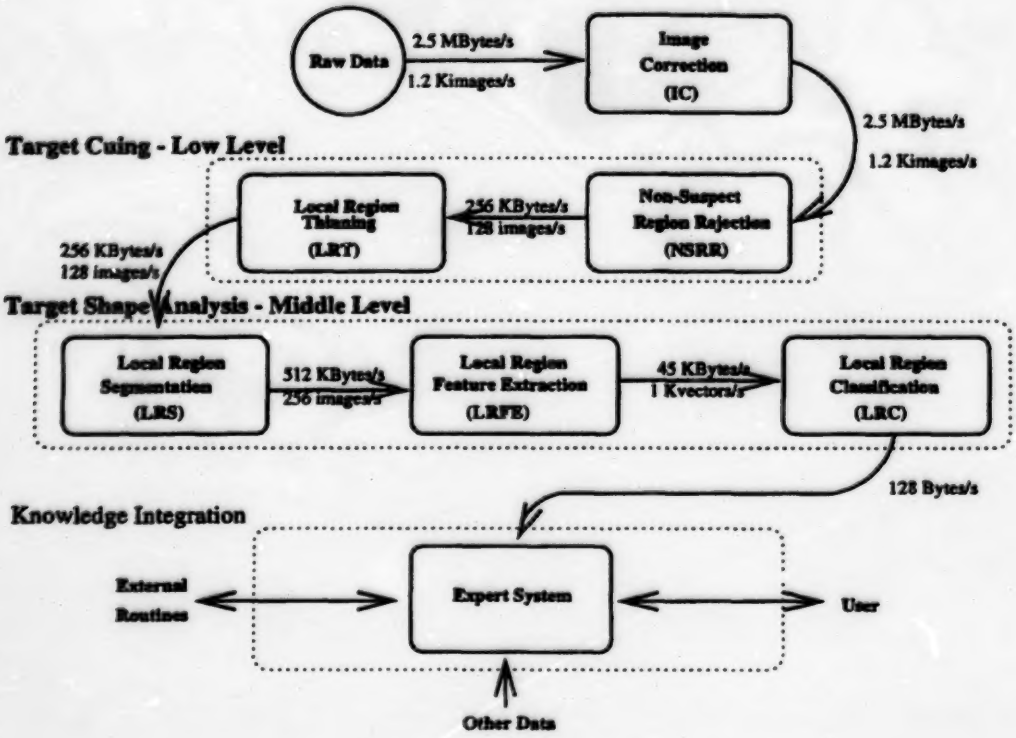



**Figure 3.3:** Systems integration analysis for FLIR imagery.

From figures 3.1, 3.2 and 3.3, the systems integration analysis for a network of transputers based on simulated images can be summarized as following:

| Function Stage | AAIR (1024x1024x8bit) | CASI (512x512x12bit) | FLIR (800x400x16bit) |
|---|---|---|---|
| Raw data input | 8 Mbytes/s, 8 frames/s (8 Ksubimages/s) | 1 Mbytes/s, 2 frames/s (512 subimages/s) | 2.5 Mbytes/s, 4 frames /s (1.2 Ksubimages/s) |
| Image Correction | 8 Mbytes/s, 8 frames/s (8 Ksubimages/s) | 1 Mbytes/s, 2 frames/s (512 Ksubimages/s) | 2.5 Mbytes/s, 4 frames /s (1.2 Ksubimages/s) |
| NSRR | 256 Kbytes/s, 256 subimages/s | 128 Kbytes/s, 64 subimages/s | 256 Kbytes/s, 128 subimages/s |
| LRT | 256 Kbytes/s, 256 subimages/s | 128 Kbytes/s, 64 subimages/s | 256 Kbytes/s, 128 subimages/s |
| LRS | 512 Kbytes/s, 512 subimages/s | 256 Kbytes/s, 128 subimages/s | 512 Kbytes/s, 256 subimages/s |
| LRFE | 45 Kbytes/s, 1 Kvectors/s | 45 Kbytes/s, 1 Kvectors/s | N/A |
| LRC | 128 bytes/s | 128 bytes/s | N/A |
| Clustering | 0.2 Kbytes/s | 0.2 Kbytes/s | N/A |
| Syntactic Feature Extraction | 0.12 Kbytes/s | 0.12 Kbytes/s | N/A |
| Statistical Feature Extraction | 0.3 Kbytes/s | 0.3 Kbytes/s | N/A |

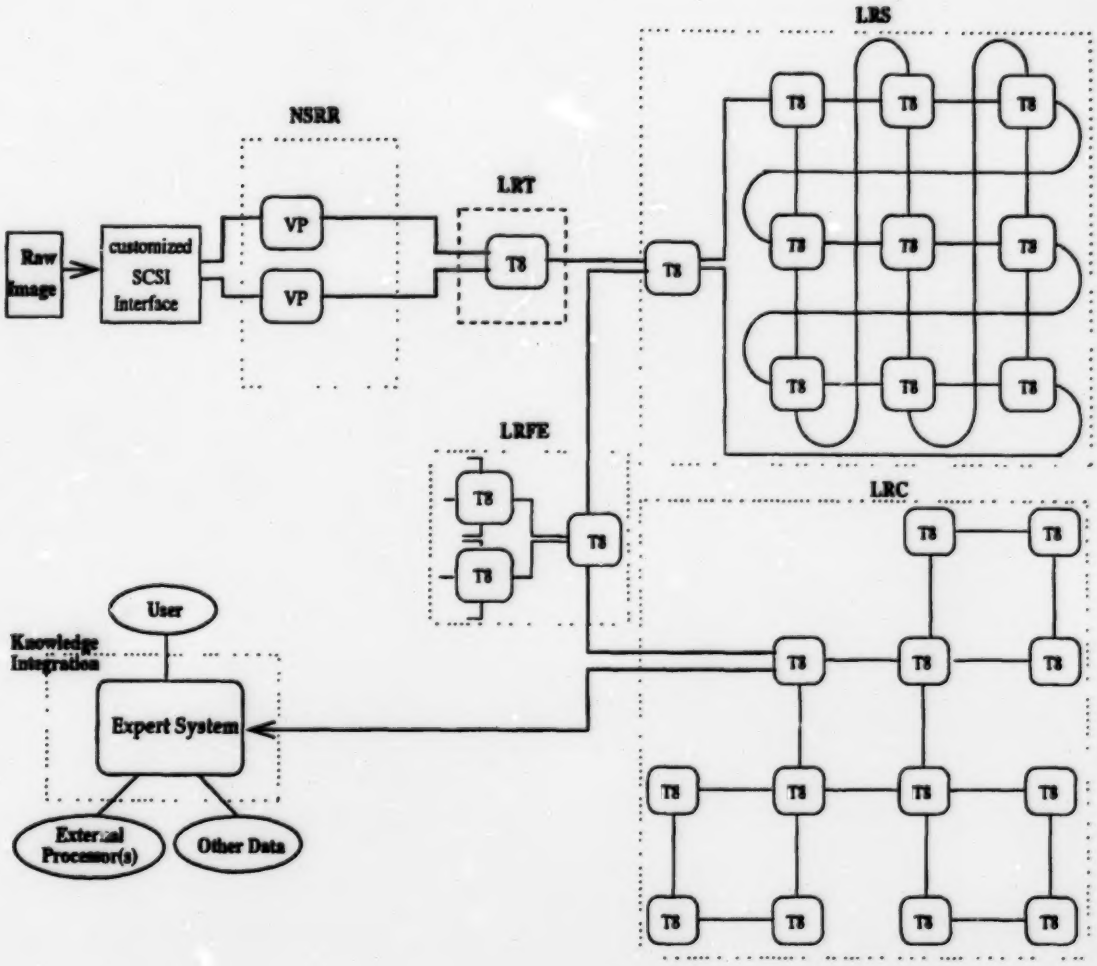
**Figure 3.4:** Data rates for a complete real-time RMD algorithm based on simulated images. Each subimage is a 32x32 pixel scan window, and threshold Iclip=32.

**3.2.2 Hardware Implementation Proposal:** In order to achieve real-time operation, a distributed computer system is essential.[3, 16] This system employs a large number of processors with varying topologies for each of the processing nodes. Thus it uses a simple point-to-point communications protocol for interconnection. The network is scaleable, i.e., increasing the number of elements requires only minor changes to the programming code. Using a cost tradeoff and data rate analysis, a parallel processing computer hardware configuration proposal for a down-scaled minefield detection system is illustrated in Figure 3.5. This system utilizes an array of transputers T800s (T8), vector processors (VP) i860s, and a Sparc station for operator interface and storage.

The INMOS transputer was selected by the TDG for this project because of its extremely high computational rate, scalability and cost.[3] A transputer is a single VLSI device with RISC processor, high-speed memory and serial communication links for direct connection to other transputers. A 32 bit CMOS T800 has a data transfer rate of 2.4 Mbytes/s. Concurrent systems can be constructed from a collection of transputer elements operating concurrently and communicating or interchanging data through links. One advantage of this system structure is that the number of transputer elements can be increased without decreasing data throughput to neighboring transputers (a problem faced by bus based systems). Also, the commonly used

transputer programming language, OCCAM, is amenable to the addition of more transputer elements without substantial modification to the existing code. Computation of processing time for NSRR, LRT and LRFE modules based on transputers is described in Appendix C. Also, to solve the i/o bottleneck problem between the SCSI bus and the RMD system input, slower T425 transputers might be employed for a customized SCSI interface.[16]

The Transtech TTM110 vector processor, an i860 based TRAM (TRAnsputer Module), has the advantages of high performance vector processing to the transputer based parallel processing architecture, and thus well suits the NSRR vectorizable algorithm.[16] This 64 bit RISC processor has a throughput of 80 Mflops/s, is much faster yet simpler to code than that of T800.




VP: i860 based vector processor. T8: T800 transputer with local memory.

**Figure 3.5:** Hardware configuration proposal for the real-time down-scaled RMD system.

During Phase 1 of this contract, several vector processors and transputers had been acquired for the algorithm implementing process. Figure 3.6 summarizes the parallel equipment required and obtained by the end of phase 1. The system also consisted of three (3) parastations for the processor interconnection, and one (1) Sparc station for working environment and storage.

| RMD Modules | Parallel Equipment | | |
|---|---|---|---|
| | i860 | T800 | T405 |
| NSRR | 2 | | |
| LRT | | 1 | |
| LRS | | 10 | |
| LRFE | | 3 | |
| LRC | | 13 | |
| **Total Requirement:** | 2 | 27 | |
| **Available:** | 2 | 29 | 1 |

Note: transputer T405 might be used in the customized SCSI Interface.

**Figure 3.6:** Hardware estimation for the down-scaled minefield detection system.

## 3.3 Preliminary Test Results on Down-Scaled System

Modules of the RMD algorithm had been developed separately and on different architectures, because the necessary hardware to completely implement the algorithm was not yet acquired. Thus the modules had not all been run as one cohesive unit on the intended (target) hardware, and only preliminary tests using single frame (non real-time) imagery had been done.

Real remote minefield detection imageries (AAIR, CASI and FLIR) were not yet available at the time of this test. Thus an image of (1024x1024 pixels, 2 bytes/pixel) generated by IDL graphics package was used instead of actual AAIR images. In Figure 3.7, a typical synthetic AAIR image shows three anti-tank mines with nearby anti-personnel mines, which were laid manually using graphic software. More details on construction of computer model minefield images are given in [3].

Also, parts of the above synthetic AAIR image were carefully cropped, accordingly to the standard sizes of CASI (512x512 pixels) and FLIR (800x400 pixels), to resemble CASI and FLIR images. These simulated images therefore cannot represent the true characteristics, such as mine shapes or picture resolutions, of their imageries; they totally bear the same features of the synthetic AAIR image except for the frame sizes. They however were used to test the down-scaled system's performance.

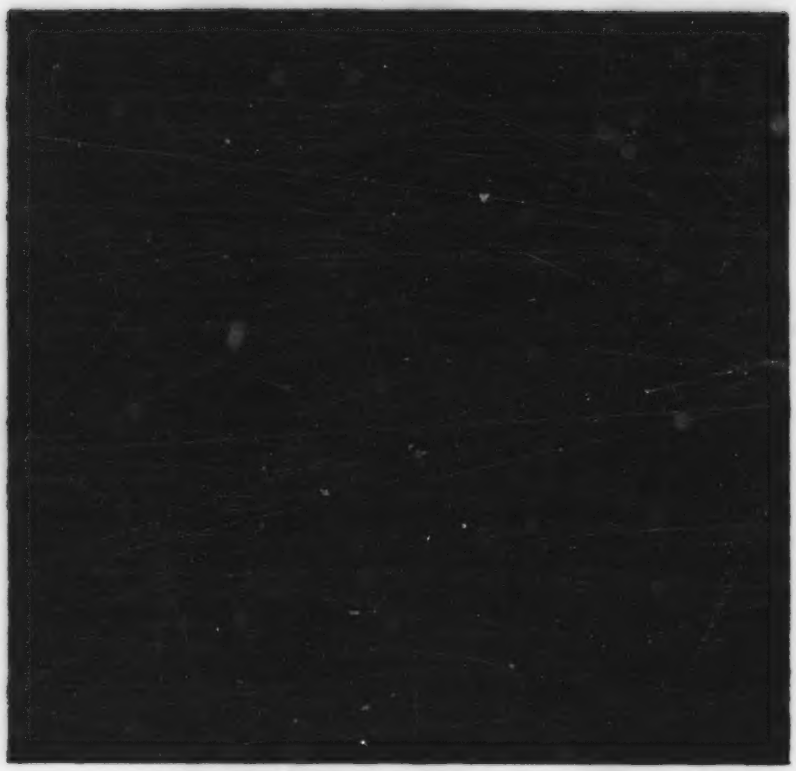


**Figure 3.7:** A synthetic AAIR image.

Figure 3.8 summarizes the types of synthetic minefield images, their standard frame sizes and image input rates (the speed of frames expected to flow into the system), and the actual measured times for reading data from local hard drive (because images were simulated and stored on hard drive instead of input from external source) in the real-time down-scaled RMD test.

| Image Type (synthetic) | Frame Size (pixel) | Image Input Rate (frames / s) | Hard Drive Reading Time (s) |
|---|---|---|---|
| AI | 1024 x 1024 | 8 | 0.362714 |
| FLIR | 800 x 400 | 4 | 0.143847 |
| CASI | 512 x 512 | 2 | 0.100918 |

**Figure 3.8:** Synthetic minefield images used for the preliminary tests.

Figure 3.9 compares the module processing times required by the systems integration analyses (discussed in section 3.2; see also [16]) with the measured times during the tests for each parallel computing hardware. For example, an airborne image size of 1024 x 1024 with data flow of 8 images/s, requires a time to process each image of 1/8 = 0.125 second for every RMD modules. The processing time for NSRR using 1 vector processor TTM110-16/4 was 0.404 second. This suggests NSRR requires four (4) TTM110-16/4 to handle airborne imagery. As a note, another way to estimate the processing times for each modules is to use internal cycle times of the processors, which is represented in Appendix C.

| Image Size (pixel) | Processing Time (s) | | | | | |
|---|---|---|---|---|---|---|
| | NSRR (1 TTM110-16/4) | | LRT (1 T805/30) | | LRS (10 T805/30) | |
| | Required | Measured | Required | Measured | Required | Measured |
| 1024 x 1024 | 0.125 | 0.404 | 0.125 | 0.209536 | 0.125 | 0.14297 |
| 800 x 400 | 0.25 | 0.095 | 0. 25 | 0.010688 | 0. 25 | 0.04926 |
| 512 x 512 | 0.5 | 0. 5 | 0.100 | 0.050368 | 0. 5 | 0.03725 |

**Figure 3.9:** Processing time and hardware configuration for NSRR, LRT and LRS in the AAIR preliminary tests.

The processing times for LRFE and LRC hardware configurations were measured in Figure 3.10. Note that these modules only process 32x32 byte subimages, presuming a synthetic AAIR image was used.

| SubImages (32x32 pixel) | Total SubRegions (28 SubRegions / subImage) | Processing Time (s) | | | |
|---|---|---|---|---|---|
| | | LRFE (3 T805/30) | | LRC 6 classes (6 805/30) | |
| | | Required | Measured | Required | Measured |
| 32 | 896 | 0.125 | 0.91168 | 0.125 | 0.97 |

**Figure 3.10:** Processing time and hardware configuration for LRFE and LRC in the AAIR preliminary tests.

# 4. Mine Detection in FLIR images

The hierarchical algorithm was initially designed for airborne imagery, and was tested using synthetic AAIR images. As real vehicle-mounted sensor images (Forward Looking Infrared, FLIR) became available, there was obviously a need to adapt the algorithm to work with this new type of image. During phase 2 and 3 of the contract, a non real-time simulation version of the algorithm was completely implemented to detect mines in real FLIR images. This work ensures that the algorithm can cope with real image characteristics, and it serves as a general guide for later adaptations for other imageries such as CASI.

## 4.1 FLIR Imagery Characteristics

FLIR images can record individual mines or groups of mines, but the whole minefield pattern can not generally be seen. Therefore the high level of RMD algorithm (Target Spatial Analysis), which is responsible for recognizing a minefield pattern, can be entirely eliminated. However, the mine shapes are usually distorted because of the aspect angle and the shallow depth of field of the camera. An additional software component may be required to restore the image perspective. But, to retain the hardware configuration proposed in Phase 1 of this contract and to achieve real-time speed, correction of this problem can be done by utilizing different thresholds and reference values for the mine feature vectors and background feature vectors. Our initial study showed that all of the 11 morphological feature components can be used effectively in the FLIR case. Furthermore, because of the slow speed of the detection vehicle, which consequently produces a lower input data rate, a smaller hardware configuration for the Target Cueing level might be needed in comparison to the AAIR case. Also, the number of mines in a FLIR image is small and the mine-like object sizes are rather large implying that less computation is likely needed to identify suspect regions.

## 4.2 Algorithm Modification and Implementation for FLIR imagery

Real FLIR image generally does not possess idealistic features such as those in simulated AAIR image, and it also has its own unique characteristics. Low signal-to-noise ratio, large object surface, great shape distortion, and different intensity contrast in each region are common FLIR properties. Hence, modifications are certainly needed for each algorithm component. A quick solution to adapt the algorithm for FLIR is to develop a non real-time software-based system and examine it thoroughly; modifications then can be made easily, and will be transferred to a real-time version in the future.

This simulation version, named "RMD-FLIR system", was fully implemented and tested on a Unix Sparc workstation. Only the low and middle levels of the algorithm are needed. The source code was stored in the TDG directory:

Programmer: S. Duong
Location: /net/frodo/export/home/projects/rmd/hieralgorithm/rmd-flir
Language: IDL version 5.1, C
Platform: Unix SparcStation

Several software tools were utilized in the development:

- Interactive Data Language (IDL): provides the main programming environment
- On-Line Pattern Analysis and Recognition System (OLPARS): analyzes data by performing pattern recognition functions
- Subroutine Package for Image Data Enhancement and Recognition (SPIDER): image analysis package which provides segmentation algorithms

The following are high lights of the programming process and modifications made on RMD hierarchical algorithm to adapt to FLIR images. More details of software development of a complete algorithm for RMD are given in "Image Analysis Application Development" report[2].

### 4.2.1 Non-Suspect Region Rejection (NSRR)

The hierarchical algorithm first needs to detect all suspect objects in an image. The Non-Suspect Region Rejection algorithm fulfills this task by comparing the mean intensity of a scan window to its local background. A high contrast indicates the existence of an object. Because there are possibly so many suspects in an image, only a preset number of objects which have the highest contrast values are considered for the mine detecting process.

At the start, grey-level data of a 800x400 FLIR image is resized into 200x100 and 100x50 subimages; these sizes by experiment work best to reveal the intensity contrast of the objects. The pixel values of these two subimages are subtracted to obtain the intensity contrast of the objects against their local background. Then NSRR scans, line by line, this 200x100 subimage to pick up the top "Iclip" suspects in each line. Some areas in a FLIR image might be very dark, so that NSRR would select these locations and thus miss the potential mine-like objects. A bigger "Iclip" value can solve this problem but however will slightly increase the number of calculations in LRT stage. Currently the "Iclip" threshold is tuned to be 30. The indices, x and y coordinates, and contrast value of these top ranked suspect spots are finally passed to the next module of the RMD system.

Some modifications were made:

- A new averaging method was introduced to better pick up the contrast between objects and background. This method also takes into account that mine-like objects can be both darker or lighter than the background, while they are only lighter than background in AAIR images.

- Some custom functions were replaced by IDL built-in routines for speed improvement. For example, the "Rebin" IDL function was used instead of "Blocksumit" to reduce image size.

- A different "Iclip" threshold (number of highest ranked values) was experimentally selected to cover most of the mine-like objects without including undesirable clutter.

- Because of larger object size in the images, a new, more appropriate, size of scan window was considered. The chief area of interest in a FLIR image is the middle strip. In that region mine shapes are relatively undistorted and large enough for shape discrimination. They are also far enough in front of other sensors to allow time for data fusion. Therefore, a scan window of 100x50 pixel was elected for best performance. This window size is big enough to cover an anti-tank or anti-personnel mine. A smaller scan window tends to falsely pickup dark regions at the top area of FLIR images.

Figure 4.1 shows a result from NSRR: tiny black dots represent the centers of scan windows where NSRR detects high intensity contrast.

**Figure 4.1:** A result from NSRR.

### 4.2.2 Local Region Thinning (LRT)

The previous NSRR module produces a list of coordinates and intensities. However, there is no guarantee that the scan windows, centered on each coordinate, will not significantly overlap with neighboring windows. Local Region Thinning prevents adjacent hits from creating many regions when one region will do.

Starting at an x-y coordinate, LRT looks around for any other x-y coordinates which fall within a "separate" limit. It then recalculates these coordinates together into a single one, which centers at the highest intensity spot. LRT then erases all the initial coordinates and passes the new centers to LRASSEMBLE function.

LRASSEMBLE is an extension of LRT module. Because LRT only outputs the coordinates of the centers of suspect objects, LRASSEMBLE will construct 64x64 grey scale subimages around these centers so that the segmenter in LRS can analyze.

A "separate" threshold value of 70 successfully combined parts of a unique object together. This threshold still ensures the non-overlap rule of the subregions. Also, a different frame size of 64x64 pixels was selected to capture the suspect regions for later shape analysis. An output from LRT and LRASSEMBLE (Figure 4.2) shows that the number of suspect objects was reduced from Figure 4.1, and these objects were framed for later processing.

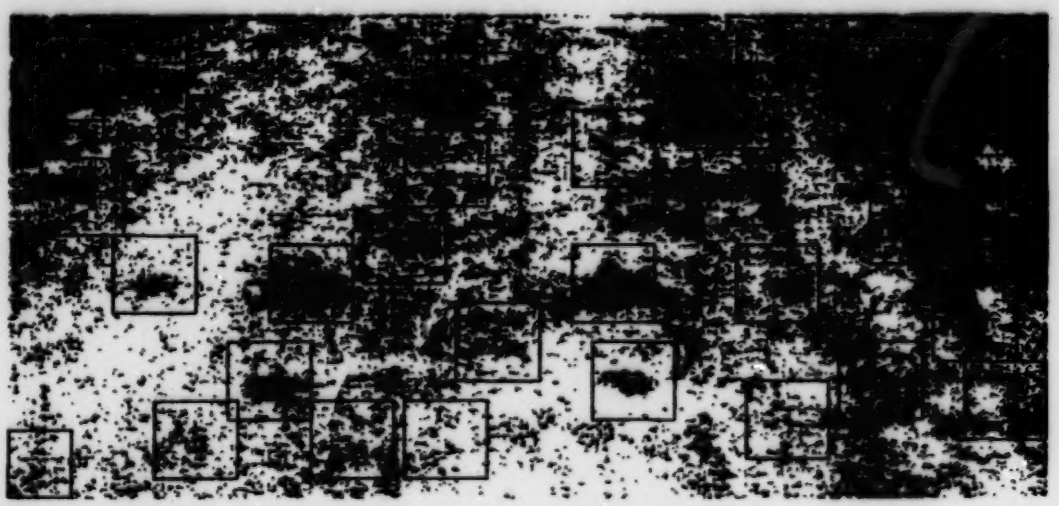


**Figure 4.2:** An output from LRT and LRASSEMBLE.

### 4.2.3 Local Region Segmentation (LRS)

Local region segmentation attempts to partition subimages into a set of uniform regions. Further reading on segmenters can be found in [2]. Follow a study in [2], only two algorithms RSAM (Region Split And Merge) and MHFM (Marr Hildreth Fill and Merge) were extensively tested for FLIR images. Finally RSAM was elected because of its speed and accuracy.

RSAM currently takes as input a 64x64 grey-level subimage and provide a 64x64 labeled subimage as output. These subimages have their centers at the x-y coordinates passed from the LRT stage. This 64x64 size of input subimage however does not guarantee to cover the entire mine shape in some cases, but RSAM strictly requires a $2^n$ frame size. Other threshold values were also carefully adjusted to yield an acceptable result as small regions have to be discarded. An optimal choice was:

- Level at which region segmentation starts, L0 = 0.
- Threshold range of intensity allowed in a region for split-and-merge evaluation, ITHV = 70.
- One-dimensional array in which threshold of difference in average intensities is stored as function of area in small region elimination, NTAREA = 9.

It is very common that practical FLIR images of mines have many tiny dots scattered around suspect objects, and these dots are usually much darker than the main targets. To save processing time for the segmenter, it is desired to clean up this clutter by using the mean of a 64x64 window as the intensity threshold to filter out all objects that are darker. Note also that in FLIR some areas are noticeably darker than the others, for example the upper left and right regions of the FLIR image in Figure 4.2. Using the mentioned window mean value will offset this undesired characteristic, because objects will be compared to local background instead of the gray value of the whole image.

As shown in Figure 4.3, RSAM sometimes does not wholly separate the main target from its surrounding blots. The reason for this is RSAM was mainly designed for objects which have smaller object size / pixel size ratio, and objects should not connect to their adjacent pieces by any narrow "bridges", which unfortunately is not the case in FLIR. To compensate this disadvantage, the IDL function Smooth was utilized to blur out those small "bridges" as much as possible, and to produce a clean smooth surrounding edge. The degree of smoothing has to be tuned so that the segmented images do not lose their characteristic shapes and still satisfy the uniform surface requirement. Some other IDL functions, such as Dilate & Erode and Leefilt, were also tried but did not yield better results.

Another important enhancement was made to improve the segmentation process and computation speed. Because a 64x64 scanned window (subimage) in LRT successfully positions the suspect object, which occupies a large fraction of the subimage, at near the window center, and LRS homogeneously smoothes out the object' surface, the FLIR variant of the algorithm therefore needs to concern only two regions in this 64x64 subimage: the suspect at center and the background (for example, see the good image of a anti-tank mine in Figure 4.3). By applying

this technique, LRFE will spend less time to extract features of the regions. Without this scheme, RSAM can produce as much as 20 regions from one 64x64 subimage, each region might have 9 to 11 features.

Scanned images of anti-tank mines Scanned images of anti-personnel mines
(in each case, a good image is at left, and a bad image with narrow "bridges" is at right)

Unknown object (clutter)

**Figure 4.3:** Mine images were segmented using RSAM algorithm.

### 4.2.4 Local Region Feature Extraction (LRFE)

A vector of feature values are next calculated for each region output by the LRS. The feature values are quantities that the classifier LRC will use to judge which objects are mine-like. The mathematical definitions of these features are in Appendix B. This external routine, Local Region Feature Extraction, is written in C as a TDG supporting program.

At the start 11 features were considered: area, mean, stdev, perim, max, min, length, width0, width1, width2, rm2. Further analysis of the feature vectors using On-line Pattern Analysis and Recognition System (OLPARS) software suggested to reduce the number of features to 9, namely, the region area (the number of pixels in the region), intensity mean, intensity variance, maximum intensity, minimum intensity, 4-adjacency perimeter (measures the length of the perimeter of the region using 4 adjacencies), 4-adjacency size (calculates the size, area/perimeter), 4-adjacency compactness (weights the compactness, $2\pi$ * area / perimeter square), and height/width ratio (a measure of how skinny the object is; the height and width are calculated from the smallest circumscribing rectangle).

When an object image has many tiny dark spots in its region, it might crash LRFE external function for some calculation overload. The Smooth function introduced in LRS (section 4.2.3)

will serve well to fill up those spots. However, too much smoothing will saturate RSAM and thus produce no segmented images.

In the FLIR algorithm, because the LRS module produces only two regions (the suspect at center of each 64x64 subimage and the background), LRFE should ignore the calculation for the background. This will save both speed and data storage.

### 4.2.5 Local Region Classification (LRC)

In the Local Region Classification stage, a classifier will use a database of feature vectors of known mines, called "design" class vectors, to test against those of an unknown object. The classifier combines the feature values of an object into one or more values which will be compared with the database.

For mine class analysis, the RMD simulation was run on a set of FLIR test images, supplied by the TDG Scientific Authority. After the LRFE step, 13 images of anti-tank mines, 27 images of anti-personnel mines and 166 background images were selected to extract their feature vectors. Using the pattern analysis tool OLPARS to analyze these vectors revealed that mine classes are clustered and well separated from background classes (Figure 4.4). Thus misclassification can ideally be minimized.

t: anti-tank data; p: anti-personnel data; J: clutter data

**Figure 4.4:** Distribution of features of anti-tank, anti-personnel mines and background

Although OLPARS suggests Fisher Pairwise Classifier has better performance (Figure 4.5), Nearest Mean Classifier (NMC, Figure 4.6) was used for the Local Region Classification in this experiment, because it is simple, fast, and already available. However, Fisher Pairwise Classifier should be used for the future FLIR algorithm.

Confusion Matrix For: FISHER PAIRWISE DESIGN
Data Set: a-tp 11/ 11
Classifier: fp
Classifier Text:
Thresholds: 1 Minimum Vote Count: 2

| | tank | person | clutter | TOT | COR | REJ | % C | % R |
|---|---|---|---|---|---|---|---|---|
| tank | 13 | 0 | 0 | 13 | 13 | 0 | 100.0 | 0.0 |
| person | 0 | 24 | 3 | 27 | 24 | 0 | 88.9 | 0.0 |
| clutter | 1 | 24 | 137 | 166 | 137 | 4 | 82.5 | 2.4 |
| | 14 | 48 | 140 | 206 | 174 | 4 | 84.5 | 1.9 |

Confusion Matrix For: OVERALL EVALUATION - FPW
Data Set: a-tp 11/ 11
Classifier: fp
Classifier Text:
Thresholds: 1 Minimum Vote Count: 1

| | tank | person | TOT | COR | REJ | % C | % R |
|---|---|---|---|---|---|---|---|
| tank | 13 | 0 | 13 | 13 | 0 | 100.0 | 0.0 |
| person | 0 | 27 | 27 | 27 | 0 | 100.0 | 0.0 |
| | 13 | 27 | 40 | 40 | 0 | 100.0 | 0.0 |

**Figure 4.5:** Fisher Pairwise Classifier evaluation.
(**tank**: anti-tank data; **person**: anti-personnel data; **clutter**: noise data)

Confusion Matrix For: Nearest Mean Classifier
Data Set: a-tp 10/ 11
Classifier: a-nm
Classifier Text:
Thresholds: 1

| | tank | person | clutter | TOT | COR | REJ | % C | % R |
|---|---|---|---|---|---|---|---|---|
| tank | 7 | 1 | 0 | 8 | 7 | 0 | 87.5 | 0.0 |
| person | 5 | 9 | 0 | 14 | 9 | 0 | 64.3 | 0.0 |
| clutter | 42 | 27 | 97 | 166 | 97 | 0 | 58.4 | 0.0 |
| | 54 | 37 | 97 | 188 | 113 | 0 | 60.1 | 0.0 |

| | tank | person | TOT | COR | REJ | % C | % R |
|---|---|---|---|---|---|---|---|
| tank | 7 | 1 | 8 | 7 | 0 | 87.5 | 0.0 |
| person | 5 | 9 | 14 | 9 | 0 | 64.3 | 0.0 |
| | 12 | 10 | 22 | 16 | 0 | 72.7 | 0.0 |

**Figure 4.6:** Nearest Mean Classifier evaluation.
(**tank**: anti-tank data; **person**: anti-personnel data; **clutter**: noise data)

An important key for a successful LRC is to design a good database so that a classifier will depend on it for identifying objects: a small database will leave many mime objects unrecognized; while an abundant database will cause the classifier to select many clutters, to be highly bias (over trained) on a certain set of test images.

Furthermore, some knowledge of designing class vectors (database) is required for classifiers. In Nearest Mean classifier case, a designed database generally starts with a small set of known mines which yield good segmented images. The LRFE algorithm is run to collect more feature vectors of many other mines. Then LRC classifies these feature vectors against the initial database. Based on the "Distance Estimation" (D.E.) results from the LRC to build a new larger database:

- Select objects that have big D.E. values because they will significantly differ from the first set of mines data.
- Those having clustered D.E. values will likely share similar characteristics, and thus choose only one object to avoid data abundance.

The current LRC has a class design base of 12 anti-tank and 12 anti-personnel mines. Also, a threshold is carefully designed and tuned to maximize the mine-like detection and minimize false alarms. Another method to improve the success rate is to put weight factors on those features which most significantly separate classes. This scaling can be established inside the LRC external routine or in the IDL main program.

## 4.3 Implementing and Testing the RMD-FLIR System:

A fast approach to prove that the hierarchical algorithm works well with real FLIR images is to implement it as a non real-time (simulation) version on a workstation. Interactive Data Language (IDL) software package is chosen as the programming environment because of its high level language and the advantage of interactive and visual computing. Together, the power of numerical analysis, image processing, array-oriented commands and graphical user interface (GUI) of IDL help to develop a full system such as the FLIR algorithm quickly and efficiently.

Real FLIR images, acquired from ILDP, were stored in AGEMA's image format and were transformed into TDG format using IDL routines. Each image frame has 800 x 400 12-bit pixels. The data input rate for real-time version is expected to be 4 frames/s. System integration analysis will be required in the real-time system to guarantee this speed.

Figure 4.7 compares the processing time for each modules of the simulated algorithm designed for real FLIR and for synthetic AAIR. The table indicates a shorter processing time for FLIR imagery because a FLIR image has smaller size and contains fewer number of suspect objects.

The test run on a SPARCstation-20. A typical FLIR (Figure 4.1) has 2 anti-tank and 4 anti-personnel mines, while a typical AAIR (Figure 3.7) has 3 anti-tank and 6 anti-personnel mines.

| **Module** (simulation) | **FLIR** (800x400 16bits) | **AAIR** (1024x1024 8bits) |
|---|---|---|
| NSRR | 0.25207007 | 4.8207611 |
| LRT | 0.31768095 | 28.030519 |
| LRASSEM | 0.26922405 | 0.72288501 |
| LRS | 4.6891500 | 6.7269039 |
| LREF | 9.0260390 | 8.3981580 |
| LRC | 6.0657340 | 10.572046 |

**Figure 4.7:** Processing time (seconds) for IDL versions of the algorithm using FLIR and AAIR images.

At the time there were 44 FLIR images available for testing, 11 of them were so poor in quality that they cannot be used to detect mine even with human vision. The FLIR algorithm was run with the remaining images and the following results were obtained:

| Image | T.P.A.* | Image | T.P.A.* |
|---|---|---|---|
| 960716_000000_0170 | 2 1 4 | 960716_041500_0160 | 2 3 5 |
| 960716_001500_0359 | 2 3 6 | 960716_043000_0370 | 2 3 5 |
| 960716_003000_0110 | 2 1 6 | 960716_044500_0060 | 1 4 5 |
| 960716_004500_0299 | 2 1 4 | 960716_050000_0250 | 1 3 6 |
| 960716_010000_0490 | 2 2 8 | 960716_051500_0460 | 2 2 6 |
| 960716_011500_0180 | 2 4 9 | 960716_053000_0160 | 2 2 5 |
| 960716_013000_0390 | 2 2 4 | 960716_054500_0350 | 1 3 4 |
| 960716_014500_0110 | 1 2 5 | 960716_060000_0540 | 2 4 6 |
| 960716_020000_0310 | 2 2 7 | 960716_061500_0249 | 1 2 3 |
| 960716_021500_0509 | 2 2 6 | 960716_063000_0450 | 1 2 3 |
| 960716_023000_0210 | 2 2 6 | 960716_064500_0169 | 1 1 3 |
| 960716_024500_0429 | 2 2 5 | 960716_070000_0361 | 2 4 7 |
| 960716_030000_0130 | 2 3 9 | 960716_071500_0549 | 2 1 4 |
| 960716_031500_0321 | 2 2 7 | 960716_073000_0260 | 1 1 2 |
| 960716_033000_0540 | 1 2 4 | 960716_074500_0449 | 2 1 4 |
| 960716_034500_0260 | 1 4 5 | 960716_080000_0140 | 2 0 3 |
| 960716_040000_0450 | 1 2 6 | | |

* **T.P.A.**: number of anti-tank mines, anti-personnel mines, and total suspect objects (hits) that FLIR algorithm found in each test image.

**Figure 4.8:** Simulated FLIR algorithm test result.

There were 2 anti-tank (AT) and 4 anti-personnel (AP) mines in each FLIR image. For 33 test results in Figure 4.8 there were 55 anti-tank and 73 anti-personnel mines found, 44 false alarms in 172 hits. Therefore the test result can be calculated as:

| Anti-Tank found | Anti-Personnel found | Success hit | False alarm |
|---|---|---|---|
| 83.3 % | 55.3 % | 64.6 % | 25.6 % |

**Figure 4.9:** Simulated FLIR algorithm success rate.

From the test result in Figure 4.9, anti-tank mines can be easily detected because their shapes are quite large and stay distinctly from the neighbors. In the other hand, anti-personnel mines are smaller and usually connected with other blobs via "bridges", thus their true shapes are altered and cannot be detected using the current classifier. Moreover, many small clutter spots combine with each other to form a structure which has feature characteristics closely resembling those of anti-personnel mines, thus yields more false alarm. Increasing the classification threshold will recognize more anti-personnel mines but will also include more clutter (false alarms). Currently there is no certain feature that can be used to distinguish anti-personnel mines and random structures. One solution for this problem is a combination of image filter and segmenter which can entirely isolate a small mine shape from its surrounding.

Another factor which decreases the rate of success: many random structures (clutter combined from smaller blots) or some particular areas of a FLIR image might have greater contrast values than those of anti-personnel mines. Because LRT only picks the top "Iclip" contrast values, some anti-personnel mines will be left unconsidered. Possible corrections might be to increase the "Iclip" threshold, to scan contrast values independently in many non-overlap regions of FLIR, or to normalize the whole image before RMD processing.

# 5. Conclusion

At the completion of the contract, the following tasks were accomplished:

1. A real-time down-scaled RMD system consisting of low and middle levels of the algorithm was completely implemented on a network of array processors and transputers. Preliminary tests with synthetic airborne images indicated the system responded as predicted in term of system integration and real-time performance.

2. A hardware proposal for a complete target system was estimated for AAIR, CASI and FLIR imageries. Several parallel processing computer hardware components were acquired to build a minimal prototype. This fulfilled the Initial Equipment Acquisition.

3. Ground based passive infrared images (FLIR) and some hyperspectral images (CASI) were acquired and stored.

4. An upgrade of the expert system software has been made. Staff training on the expert system was also achieved.

5. Adaptation of the RMD hierarchical algorithm for FLIR was studied and necessary modifications were made. This included tuning the system parameters, and developing new techniques to produce necessary outcomes. Real FLIR images were converted to be ready for RMD processing.

6. A non real-time version of the hierarchical algorithm was fully developed to run with real FLIR imagery. This system was coded using IDL language and C as external routines, and ran on a Unix-SparcStation platform. The FLIR algorithm was tested using real FLIR images. The results were reasonably satisfactory with 64.6% probability of detection and 25.6% probability of false alarm. Many suggestions for future enhancement of the FLIR algorithm were also given.

7. By building a simulation version of the FLIR algorithm, it proves that it worked well with real image data. All modifications then can be transferred to a real-time version. Another benefit is one can gain experience to adapt the algorithm for other types of imageries.

Future work should search for a better segmenter, investigate the process to put weighting factors for each minelike class, and fine tune the overall system. Then the whole system can be implemented on parallel hardware or some new fast computing platform to achieve real-time performance.

The next logical step is to adapt this RMD algorithm for CASI, both in simulation and in real-time. Since CASI does not require target shape analysis but spatial analysis, more research on

clustering algorithms is essential. And because such an analysis will require advanced computing techniques, a revision of expert system and knowledge base will possibly be a necessity.

# Appendix A: Expert System and Its Applicability

A more detailed introduction to the Expert System was given in [16]. The following are high lights of that report and some recent updates.

An expert system is a computer system, consisting of software and hardware, that attempts to mimic a human expert's thought process to solve complex problems or make decisions in a given field.

The Syntactic Pattern Recognition task of the hierarchical algorithm is an if-then-else based decision process, the type of reasoning that an expert system excels at. Therefore it is best to implement this module using expert system technology, and this unit will interact with the Syntactic Feature Extraction module for decision making. Furthermore, the top level of the RMD hierarchy involves the integration of disparate types of knowledge, including conflict resolution and whole minefield recognition. The characteristics of an expert system make it ideally suited for this task.

## A.1 Expert System Shell

A suitable commercial expert system shell, *Elements Expert* from Neuron Data Co., has been chosen to use with the RMD system because of the following significant features:

- A complete Rule-based development environment: One can quickly produce rules using a knowledge base, and this base can be updated at any given time. It can also be used as a rule server to provide services for existing applications.
- Event driven: It can work with external messages and react to them. These messages can come from an external event (such as a change to the GUI) or from an internal event (such as an action triggered by a rule).
- Problem solving strategy: Forward and backward chaining are available which can be switched on and off by choice or can be used in combination. LHS and RHS models are as well available.
- It can easily integrate with other language modules like C/C++ and Java. It also supports networking and standard relational databases, and has a cross-platform graphic user interface (GUI) builder.
- It has a friendly user interface in Unix or windows operating systems. The company provides good technical support, and even custom development contracts.

An upgrade to *Elements Expert* package and staff training were purchased in phase 2. The software is intended to run on Unix platforms and can be linked to C/C++ modules.

## A.2 Scope of the Knowledge Base

The knowledge base of an expert system is the collection of information which can be extracted by the expert system in pursuit of a solution to a problem. The significant feature of a knowledge base is that the knowledge base contains not only static data as in database but also relational information.

The static database consists of information derived from:

1. Physical geography of the suspect area:

- Types of terrain: rocks, marshes, swamps, bogs, flat uplands, river, desert, ...
- Types of vegetation: tree, shrub, herb, moss, ..., and some categories such as height.
- Texture of terrain: coarse, fine, medium, ultra-fine.

2. Tactical Command Control and Communication System (TCCCS): a distributed processing computer network system which consists of a large number of cells deployed throughout the corps area:

- Types of mine delivery means available from enemy forces: hand, mechanical dispenser, helicopter, artillery, ...
- Enemy's tactics such as supporting both offensive maneuver (flank minefields) and the defence.
- Types and availability of mines from enemy forces: anti-tank (wooden, plastic, ...) and anti-personnel mines (PMN6, PFM-1, ...)
- Position of enemy's tactical group and weapons (armor, artillery), and the method of deployment to cover the minefield.
- The threat from enemy air attack.
- The ability of the enemy's detection and command control and communication systems to execute an effective minefield deployment.
- Location and status of all friendly forces in the vicinity.

In addition, the database includes site specific information supplied from the battle field personnel or head quarters command level which can be added to or written over the knowledge base before or during processing.

Relational information includes production rules If-Then or If-Then-ELse

## A.3 Testing Methodology

Testing how well the expert system and the knowledge base perform is very crucial at this stage. The evaluation can be done by executing several test scenarios that the knowledge engineer (or expert) has already solved to see if the expert system generates the same answers through the same reasoning process.

The evaluation can be subdivided into two areas: static and dynamic:

1. The *static evaluation* is the testing of the knowledge base for its consistency and completeness. To test the completeness we should constantly ask ourselves, throughout the evaluation process, does more knowledge need to be added to produce the correct answers.

2. The *dynamic evaluation* is the testing of the reasoning process and the advice given. The accuracy and the reliability of the decision is crucial. The important part of the dynamic evaluation is asking for an audit trail of a decision and comparing it against an expert's reasoning.

# Appendix B: Feature Selection for Object Identification

$F_j^i$ is the $i^{th}$ feature of the $j^{th}$ region, and $I_j(x, y)$ is the grey level intensity at location (x, y) of the $j^{th}$ region.

- **Area:** calculates the number of pixels in the region (area)

$$F_j^1 = \sum_{(x,y) \in j} 1$$

- **Intensity mean:** calculates the grey level mean of the pixel intensities in the region

$$F_j^2 = \frac{1}{F_j^1} \sum_{(x,y) \in j} I_j(x, y)$$

$I_j(x, y)$ is the grey level intensity at location (x, y) of the $j^{th}$ region

- **Intensity variance:** calculates the grey level variance of the pixel intensities in the region

$$F_j^3 = \frac{1}{F_j^1 - 1} \sum_{(x,y) \in j} ( I_j(x, y) - F_j^2 )^2$$

- **Maximum intensity:** finds the value of the maximum grey level in the region

$$F_j^4 = \text{MAX} [ I_j(x, y) ], \quad \text{for } (x, y) \in j$$

- **Minimum intensity:** finds the value of the minimum grey level in the region

$$F_j^5 = \text{MIN} [ I_j(x, y) ], \quad \text{for } (x, y) \in j$$

- **Perimeter:** measures the length of the perimeter of the region using either 4-adjacency

$F_j^6$ = 4-adjacency length of $j^{th}$ region's perimeter
$F_j^7$ = 8-adjacency length of $j^{th}$ region's perimeter

- **Size:** calculates the size of the region

$$F_j^8 = 2\frac{F_j^1}{F_j^6} \qquad F_j^9 = 2\frac{F_j^1}{F_j^7}$$

• **Compactness:** calculates the compactness of the region. Note: the scaling factor of $4\pi$ can be omitted when used as a feature vector component

$$F_j^{10} = 4\pi\frac{F_j^1}{(F_j^6)^2} \qquad F_j^{11} = 4\pi\frac{F_j^1}{(F_j^7)^2}$$

• **Center of gravity:** locates the coordinates of the center of gravity of the region (cgx and cgy)

$$F_j^{12} = \frac{1}{F_j^{14}} \sum_{(x,y)\in j} x\, I_j(x,y) \qquad F_j^{13} = \frac{1}{F_j^{14}} \sum_{(x,y)\in j} y\, I_j(x,y)$$

• **Central moments of the region:** 0th central moment ($m_0$), 1st central moment ($m_{10}$, $m_{01}$, $m_{11}$), 2nd central moment ($m_{20}$, $m_{02}$, $m_{22}$), 3rd central moment ($m_{30}$, $m_{03}$, $m_{33}$)

$$F_j^{14} = \sum_{(x,y)\in j} I_j(x,y)$$

$$F_j^{15} = \sum_{(x,y)\in j} I_j(x,y)\,(x - F_j^{12}) \qquad F_j^{16} = \sum_{(x,y)\in j} I_j(x,y)\,(y - F_j^{13})$$

$$F_j^{17} = \sum_{(x,y)\in j} I_j(x,y)\,(x - F_j^{12})\,(y - F_j^{13})$$

$$F_j^{18} = \sum_{(x,y)\in j} I_j(x,y)\,(x - F_j^{12})^2 \qquad F_j^{19} = \sum_{(x,y)\in j} I_j(x,y)\,(y - F_j^{13})^2$$

$$F_j^{20} = \sum_{(x,y)\in j} I_j(x,y)\,(x - F_j^{12})^2\,(y - F_j^{13})^2$$

$$F_j^{21} = \sum_{(x,y)\in j} I_j(x,y)\,(x - F_j^{12})^3 \qquad F_j^{22} = \sum_{(x,y)\in j} I_j(x,y)\,(y - F_j^{13})^3$$

$$F_j^{23} = \sum_{(x,y) \in j} I_j(x,y)\,(x - F_j^{12})^3\,(y - F_j^{13})^3$$

• **Height / width:** a measure of how skinny the object is (skin). The height and width are calculated from the smallest circumscribing rectangle

$$F_j^{26} = \frac{F_j^{25}}{F_j^{24}}, \qquad F_j^{24} = \text{Width}, \quad F_j^{25} = \text{Height}$$

Note that $F_j^7$, $F_j^9$, $F_j^{11}$, $F_j^{12}$, $F_j^{13}$, $F_j^{15}$, $F_j^{16}$, $F_j^{18}$, $F_j^{19}$, $F_j^{21}$, and $F_j^{22}$ are not useful for mine identification [2].

# Appendix C: Computation of processing time for NSRR, LRT and LRFE

In the process of estimating the run time of three modules: NSRR, LRT and LRFE, six functions which govern the processing time are addition, subtraction, division, multiplication, comparison and iteration control (DO loop). Before presenting the computation of processing time of the modules, let's define the following quantities:

- $n$ = edge size of an input image. $n = 2^{(5+m)}$ where $m = 0, 1, 2, \ldots$ M is the number of times required to compress an $n$ by $n$ image to a 32 by 32 region.
- Hit : Number of mines assumed per 32 by 1024 pixels area.
- T1 = constant determined by the type of transputer used (e.g. T425, T800 or T9000).

We consider two types of transputers: T800 and T9000. The T800 is a 32 bit CMOS microprocessor and has a data transfer rate of 2.4 Mbytes/s. Its internal cycle time is 33 ns and the numbers of processor cycles performed by the computing functions: (+/-), (x), (÷), comparison, Do Loop are 7, 11, 17, 7 and 15.1 respectively [18]. The T9000 is also a 32 bit CMOS microprocessor but has much better performance. It is capable of transferring data up to 80 Mbytes/s. Its internal cycle time is 20 ns and the numbers of processor cycles performed by the computing functions: (+/-), (x), (÷), comparison, and Do Loop are 2, 2, 8, 2, and 3 respectively [19].

## C.1 Computation of processing time of the NSRR module

The NSRR program consists of two subroutines: Rebin and Compress, and a main routine. The Compress subroutine was designed to compress the samples of image via consecutive block summations, and the Rebin subroutine was made to resize images. The main routine comprises three parts: 1) Compressing the samples of image, 2) Computing the mean of all 4x4 and 32x32 windows, and 3) Locating the points of maximum contrast.

The number of computations for the two subroutines Rebin and Compress can be found in Figure C.1, for the main NSRR routine in Figure C.2, and for the entire module NSRR is in Figure C.3.

| | REBIN | COMPRESS |
|---|---|---|
| + | $128\ (n/32)^2$ | $3/2\ n^2$ |
| x | $128\ (n/32)^2$ | $3/2\ n^2$ |
| loop | $73T\ (n/32)^2 + nT/32$ | $T\ (n/2)\ (1+3n/2)$ |

**Figure C.1:** Computations for REBIN and COMPRESS subroutines.

| | | | |
|---|---|---|---|
| Main routine | Compress an n by n image to a 32 by 32 region | + | $3/8\ n^2\ (\sum_{m=0}^{M} 1 \div 2^m)$ |
| | | x | $3/8\ n^2\ (\sum_{m=0}^{M} 1 \div 2^m)$ |
| | | Do loop | $T(n/4)(1+3n/4)(\sum_{m=0}^{M} 1 \div 2^m)$ |
| | Compute the mean of the 4 by 4 and 32 by 32 regions | +/- | $128\ (n/32)^2 + (n/2)^2$ |
| | | / | $128\ (n/32)^2$ |
| | | x | $(n/2)^2 + (n/32)^2$ |
| | | Do loop | $73T(n/32)^2 + nT/32$ |
| | Locate the point of maximum contrast | + | $2(n/2)^2 + 3(n/32)$ |
| | | Do loop | $(nT / 2^M)\ [1 + 9Hit + 8Hit\ (n / 2^{M-3})]$ |
| | | Compare | $n/2\ (n/32)^3$ |

**Figure C.2:** Number of computations required by the NSRR main routine.

| | |
|---|---|
| +/- | $n^2 [ 3/8 ( \sum_{m=0}^{M} 1 \div 2^m ) + 992/1024 ]$ |
| x | $n^2 [ 3/8 ( \sum_{m=0}^{M} 1 \div 2^m ) + 257/1024 ]$ |
| / | $128 (n/32)^2$ |
| Do loop | $T [ ( \sum_{m=0}^{M} 1 \div 2^m ) (n/4)(1+3n/4) + 73(n/32)^2 + n/32 + (n \div 2^M)(1 + 9Hit + 8Hit (n \div 2^{M-3}) ]$ |
| Compare | $n/2 (n/32)^3$ |

**Figure C.3:** Total number of computations required by NSRR module.

In order to have a 2K by 4K byte minefield image processed within one second in the NSRR stage, it was found that 40 T800 or 9 T9000 transputers are needed and the maximum size of input images to each transputer is 512 by 512 when T800s are used or 1024 by 1024 when T9000s used. The data input time of a 512 by 512 image to a T800 is about .1 second and of a 1024 by 1024 image to a T9000 is approximately .01 second. The processing time (including the data input time) is presented in Figure C.4.

## C.2 Computation of processing time of the LRT module

The program for LRT is quite small and simple. Most of the computations take place in the process of calculating the distances between points in the data array to eliminate invalid candidates. The process can be easily identified in the LRT program.

As we see from Figure C.5, the processing time of LRT is a function of Iclip (Iclip is the number of candidates from the top of the output list from NSRR) and two constants: T and T1 which are determined by the type of transputers used.

| Transputer type | Image's size (byte) | Number of Hits per 32x1024 region | Processing time (sec) | Number of transputers |
|---|---|---|---|---|
| T800 | 512 x 512 | 32 | ~.87 | 1 |
| | 2K x 4K | 32 | ~1 | 40 |
| T900 | 1024 x 1024 | 32 | ~.78 | 1 |
| | 2K x 4K | 32 | ~1 | 9 |

**Figure C.4:** Processing time and number of transputers needed for NSRR.

| +/- | Iclip (3 Iclip + 4)/2 |
|---|---|
| x | Iclip (Iclip + 1)/2 |
| Compare | 2 Iclip (Iclip + 1)/2 |
| Do loop | T (Iclip) |
| While loop | T1 (Iclip (Iclip+1)/2 + Iclip) |

**Figure C.5:** Number of computations required by LRT.

If Iclip is set to be 32 (as in our application) and if an 8 Mbytes image is to be analyzed, then the number of cycles performed by one T800 transputer is 1189018. Hence, the processing time is

$$1189018 \times 33 \times 10^{-9} \approx .04 \text{ second}$$

Note: For a T800, T1 = 12 [ref. 18]

With a data transfer rate of 2.4 Mbytes/s, a T800 can import 128 Kbytes data (see Figure 3.2) in about .013 second. In short, the total run time is approximately 0.06 second, well within the required time (1 second). Thus, 1 T800 is sufficient to handle the load in the LRT stage.

## C.3 Computation of processing time of the LRFE module

The length of code of the LRFE program is long because it is composed of a number of processes ($\geq$ 10) used to compute feature components of a vector. However, each individual process is short, straightforward and independent of the others. Hence, the number of computations performed by each process can be easily calculated. A summary of the number of computations performed in the LRFE program is presented in Figure C.6.

| Feature | +/- | X | ÷ | Compare | Do loop |
|---|---|---|---|---|---|
| Mean | $\sum_{1}^{\wedge}$ | | | | T (32 x 33) |
| Area | $\sum_{1}^{\wedge}$ | | | | |
| Standard deviation | $\sum_{1}^{\wedge}$ | $\sum_{1}^{\wedge}$ | 2 | | |
| Perimeter | | P | | $13 \sum_{1}^{\wedge}$ | |
| Maximum | | | | $\sum_{1}^{\wedge}$ | |
| Minimum | | | | $\sum_{1}^{\wedge}$ | |
| Centroid | $2 \sum_{1}^{\wedge}$ | | | | |
| Length | 13/8 P(P+1) | P(P+1) | | ¾ P(P+1) | ½ T(3P+1) |
| Width | 8P | 6P | P | 4P | TP |
| 2nd moment | $\sum_{1}^{\wedge}$ | $\sum_{1}^{\wedge}$ | | $\sum_{1}^{\wedge}$ | T (32 x 33) |

A and P are the area and perimeter of a region respectively.

**Figure C.6:** Number of computations required by the LRFE module.

If we assume in the worst case that all Iclip subimages are passed to LRFE from LRS and that each subimage has an equal number *N* of square regions to be analyzed, then the total number of the regions to be processed is *N* x Iclip.

In our particular application, if LRFE module only processes regions whose sizes are at least 6 x 6 or 36 pixels (large enough to cover small mines), then for the worst case, a maximum of 28 regions in each 32 x 32 subimage are to be analyzed, or 28 x Iclip regions in total to be studied by LRFE. If Iclip is preset at 32, then the time needed to process all the regions by a 50 Mhz T9000 is ≈ 0.126 second, and by a 30 Mhz T800 is ≈ 0.8 second, within the required time (1 second).

In summary, the processing time at LRFE is a function of Iclip and the minimum size of a region chosen for the analysis. If Iclip and the minimum region size are 32 and 36 respectively, 1 T800 is sufficient to handle the load. If more than one T800 are needed, a process farm configuration is suggested.

# Appendix D: Running FLIR Hierarchical Algorithm Simulation

The following is a sample FLIR Hierarchical Algorithm simulation session. Brief descriptions and displays will be given for each stage of the low and middle levels of the hierarchical algorithm. User's guide and notes can be found in "/net/frodo/export/home/projects/rmd/hieralgorithm/rmd-flir/readme.txt".

After changing directory into "/net/frodo/export/home/projects/rmd/hieralgorithm/rmd-flir/demo", start IDL and run script file "rmd.pro". The console window displays:

```
IDL> @rmd
% Compiled module: DAT2TDG.
% Compiled module: DISPLAY_RESULT.
% Compiled module: WRITEMAPFILE.
% Compiled module: WRITETDGFILE.
% Compiled module: DECODEIMAGE.
% Compiled module: DECODEARRAY.
% Compiled module: READOBJECT.
% Compiled module: CREATEOBJECT.
% Compiled module: INITIALIZE.
% Compiled module: LOWRMD.
% Compiled module: LRASSEMBLE.
% Compiled module: LRT.
% Compiled module: MINE_CLASSIFY.
% Compiled module: NSRR.
% Compiled module: RMD_EVENT.
% Compiled module: RMD_PROCESS.
```

**Figure D.1:** Control widget.

Click "Load image" button of the above Control widget to load a 800x400 FLIR image; the selected FLIR image will be displayed:

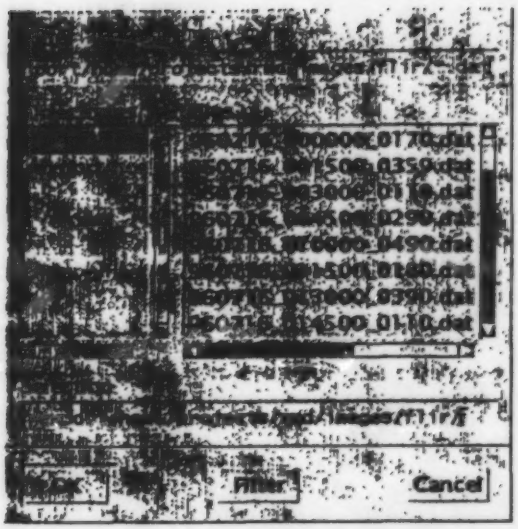



**Figure D.2:** Select FLIR image file dialog.

**Figure D.3:** A FLIR picture with 2 vivid anti-tank and 4 anti-personnel mines.

Select the Control's "Run RMD" button to start algorithm processing. For fast operating procedure, intermediate displays resulted from NSRR, LRT and LRASSEMBLE were disabled; samples of these displays are given in Figure 4.1 and 4.2. The console window shows processing times in each of the modules; this timing was done on a Sparc station 20. Then after finishing LRC, it reports the number of Hits (objects that the algorithm presumes are mines) and Suspects (objects that were suspected by the LRT but have classification distances greater than the threshold from any known class):

```
*****************************
* Low Level RMD processing *
*****************************
* Processing: NSRR
        done in      0.25207007
* Processing: LRT
        done in      0.31768095
* Processing: LRASSEMBLE
        done in      0.26922405
* Processing: LRS
        done in      4.6891500
* Processing: LRFE
        done in      9.0260390
* Processing: LRC
        done in      6.0657340

***************
* RMD result *
***************
Hits, Suspects =     4       31
```

A RMD Result window appears to show the Hits in red squares and the Suspects in black squares. Left mouse click on any of these squares will display the segmented picture of the object inside along with its feature vector; while the console window reports more information of that object such as Classification data (**Class**: classification of an object according to LRC; **LRC Dist**: measurement of how much an object's feature vector is different from the closest value in LRC database), **Position** (coordinates x and y of an object in an image frame), **Band** (the RMD result is an array of classified data of every suspect objects that were segmented in LRS, and "band" is the index pointing to an object's data within this result array), and **Contrast** (the difference between the intensity value of an object and the regional intensity mean value).

```
Select a region for information (right mouse to stop):

Class         = Anti-Tank
X,Y position  =   760   256
Band, Contrast =   27    16.1406
LRC Dist =    0.0169823

Class         = Anti-Personnel
X,Y position  =   304   192
Band, Contrast =   17    23.4844
LRC Dist =    0.0140105

Class         = Unknown
X,Y position  =   592   224
Band, Contrast =    9    33.0938
LRC Dist =    0.0347126
```

* End of RMD result

**Figure D.4:** Display of a FLIR algorithm result, an object's segmented image and its feature vector.

# References

[1] J.E. McFee, K.L. Russell, M.R. Ito, G.C. Stuart and Y. Das, "A Hierarchical Scheme for Analysis of Minefield Images", In *Proc. CRAD Signal Processing Symposium*, Defence Research Establishment Valcartier, PQ, Canada, June 1988.

[2] K.L. Russell and M.R. Ito, "Image Analysis Application Development - Final Report", DRES Contract Report 3/90, DSS Contract File No. 01SG.97702-6-2795-A, University of British Columbia, Oct. 1989.

[3] Richard C.Q. Vu, Kevin L. Russell, and M.R. Ito, "Transputer Implementation of Minefield Image Analysis - Final Report", DRES Contract Report 93-11, Contract DSS file W7702-9-R121/01-XSG, University of British Columbia, March 1993.

[4] Steven B. Achal, John E. McFee, Clifford D. Anger, "Identification of Surface-Laid Mines by Classification of Compact Airborne Spectrographic Imager (CASI) Reflectance Spectra", *Proc. SPIE Conference on Detection Technologies for Mines and Mine-like Targets, Vol. 2496*, Orlando, FL, USA, 17-21 April, 1995.

[5] Jean-Robert Simard, Jean Dumas, L'eandre S'evigny, "Improved Landmine Detection Capability (ILDC): Evaluation of the Passive LWIR Imaging Technology to Detect Low-Metal Buried Land Mines", *Proc. of NATO IRIS Conference*, London, UK, June 1996.

[6] C.G. Rey and M.R. Ito, "Image Analysis Software Development - Final Report", DRES Contract Report 40/87, DSS Contract File No. 01SG.97702-R-4-9709, University of British Columbia , April 1987.

[7] R.S.Cok., *Parallel Programs for the Transputer*, Prentice Hall, New Jersey, 1991.

[8] Andrew S.Tanenbaum, *Computer Networks*, Prentice Hall, Inc. New Jersey, 1981.

[9] G.H. Ball and D.J. Hall, "Isodata, an Iterative Method of Multivariate Analysis and Pattern Recognition", *Proc. of the IFIPS Congress.*, 1965.

[10] E. Ruspini, "A New Approach to Clustering." *Inform. Contr,. Vol. 15*, pages 22-32, 1969.

[11] I. Gath and A.B. Geva, "Unsupervised Optimal Fuzzy Clustering." *IEEE Trans. Pattern Anal. Machine Intell.* PAMI-11, pages 773-781, 1989.

[12] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum, New York. 1981.

[13] K. Rose, E. Gurewitz and G.C. Fox, "A Deterministic Annealing Approach to Clustering." *Pattern Recog. Lett., Vol. 11*, pages 589-594, 1990.

[14] F. R. Fromm and R. A. Northouse, "Some Results On Non-Parametric Clustering of Large Data Problems." In *Proc. 1st Intnl. Jt. Conf. Pattern Recognition*, pages 18-21, 1973.

[15] Yiu-fai Wong, "Clustering Data by Melting." In *Neural Computation, Vol. 5*, pages 89-104, 1993.

[16] R.C.Q. Vu and Mabo R. Ito, "System Integration Study of a Hierarchical Minefield Image Analysis Algorithm - Final Report", DRES Contract Report, DSS file W7702-1-R286/01-XSG, University of British Columbia, September 1995.

[17] J. McFee, K. Russell, M. Ito and R. Vu, "An Algorithm for Real-Time Detection of Minefields in Monochromatic Airborne Imagery", *Proc. SPIE Conference on Detection Technologies for Mines and Mine-like Targets, Vol. 2496*, Orlando, FL, USA, 17-21 April, 1995, pp. 594-605.

[18] J. McFee, S. Achal and C. Anger, "Scatterable Mine Detection Using a CASI", *Proc. of the First International Airborne Remote Sensing Conference - Volume 1*, Strasbourg, France, 11-15 September, 1994, pp. 587-598 .

[19] SGC-THOMSON Microelectronics Group. *The Transputer Databook*, 2nd edition, 1989.

[20] SGC-THOMSON Microelectronics Group. *The T9000 Transputer Products Overview Manual*, 1st. edition, 1991.